IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Provisional Application of

Title of the Invention : SDIO CONTROLLER
Provisional Application No. : 60/441,133
Provisional Filing Date : 01/21/2003
for :

VERIFICATION OF TRANSLATION

Honorable Commissioner of Patents and Trademarks
Washington, D.C. 20231
Sir:

Hiroyuki Sasai, residing at ShinjukuHiroseBldg. 4-7 Yotsuya, Shinjuku-ku, Tokyo

declares:

(1) that he knows well both the Japanese and English languages;

(2) that he translated the above-identified Provisional Application from Japanese to English;

(3) that the attached English translation is a true and correct translation of the above-identified Provisional Application to the best of his knowledge and belief; and

(4) that all statements made of his own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC 1001, and that such false statements may jeopardized the validity of the application or any patent issuing thereof.

Sunflare Co., Ltd
President
Hiroyuki Sasai

Date 18 April 2003

ENGLISH LANGUAGE TRANSLATION OF
THE NON-ENGLISH LANGUAGE PROVISONAL APPLICATTION
(37CFR1.78(a)(5))

Title of the invention
「 SDIO CONTROLLER 」

Inventor's name
Jun Takinosawa, Hiroyuki Yasoshima

Filing date of the non-English provisional application
January 21, 2003

Application No. 60/441,133

[Title of the Document] Specification

[Title of the Invention] SDIO Controller

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

The present invention relates to the SDIO controller which is a semiconductor device for connecting a SDIO host and SDIO card applications compliant with the SDIO (Secure Digital Input Output) standard, via a SD bus.

[0002]

[Prior Art]

There are a variety of standards for IC cards which can be mounted on mobile devices such as notebook personal computers (PC), personal digital assistants (PDA), digital video devices, digital cameras, and portable audio devices (these devices may be referred to collectively as "host devices"). Among others, SD memory cards have been attracting attention because of their small size, high data transfer rates, and enhanced security features.

[0003]

Today, there are two international standards for SD-related devices: SD memory standard for memory devices and SDIO standard for input/output devices (I/O). The SDIO standard is an extension of the SD memory card standard and covers input/output functions as well as memory functions. Card-shaped peripherals compliant with the SDIO standard are referred to as SDIO cards.

[0004]

A SDIO controller implements functions needed for peripherals to comply with the SDIO standard and connect to the host devices.

[0005]

The "SDIO wireless communication card" which contains the SDIO controller equipped with a UART interface and wireless communication capabilities compliant with Bluetooth is known as a conventional technology for SD-related devices (see, for example patent document 1). The SDIO wireless communication card has capabilities to communicate data from a host device wirelessly via SDIO to the distant Bluetooth-compatible wireless communication devices.

[0006]

[Patent document 1]

Japanese Patent Laid-Open No. 2002-171303

[0007]

[Problems to be Solved by the Invention]

SDIO is a relatively new standard. Besides, research and development must be conducted in such a way as to meet the specifications of the SDIO international standard.

[0008]

Environments for SDIO-related design and development are not in place: necessary hardware devices for SD interfaces and associated software are not available.

[0009]

For example, a converter between PCMCIA and SD interface is required for the existing wireless LAN module

equipped with a PCMCIA interface when developing a SDIO card provided with wireless LAN capabilities and the card is capable of mounting on the host devices equipped with a SDIO slot.

[0010]

In another example, when developing a SDIO card with large-capacity and non-volatile memory functions, the SDIO card must be equipped with a memory interface which supports various types of memory.

[0011]

The present invention is, in view of the above, to improve the design and development environments for SDIO-related devices. Its objective is to provide a highly versatile SDIO controller equipped with capabilities to interface a SDIO host with various applications (wireless, GPS, memory, etc.) as well as to provide other devices resulting from application of the SDIO controller.

[0012]

[Means for Solving the Problems]

The present invention provides a SDIO controller which is a single-chip semiconductor device connecting a SDIO-compliant SDIO host with a plurality of applications via a SD bus, comprising: a SD interface which deciphers commands received from the SDIO host and returns a necessary response to the SDIO host; one or more application interfaces; and a temporary (R/W FIFO) memory mounted between the SD interface and the application interfaces.

[0013]

The present invention makes it possible not only to select applications to use, but also to shorten the development cost and time of a card system fully compatible with the complicated SDIO standard.

[0014]

Preferably the application interfaces contain at least one of a PCMCIA, PC card bus interface, UART interface, or memory interface.

[0015]

If these interfaces are provided, SDIO cards can be developed by simply connecting to the SD controller without modifying application modules with these interfaces.

[0016]

In particular, for example, if a PCMCIA interface is provided, a SDIO card compatible with IEEE802.11b, one of wireless communications standards, can be developed by simply connecting the IEEE802.11b LAN module to the SDIO controller via the PCMCIA interface.

[0017]

Similarly, if a UART interface is provided, a Bluetooth-compatible SDIO card can be developed by connecting a Bluetooth module to the SDIO controller via the UART interface.

[0018]

Similarly, if a memory interface is provided, a memory device can be mounted, thereby adding capability of the memory device to the input/output capabilities of SDIO.

[0019]

In that case, the temporary memory preferably comprises: as many read memories (RFIFO) as the application interfaces require or at least one read memory to hold data temporarily before read out of SDIO applications; and at least one write memory (WFIFO) which temporarily holds data to be sent out to the SDIO host.

[0020]

The use of a plurality of read memories (RFIFO) makes it possible to manage data easily even if a plurality of applications are operated simultaneously and simplifying and speeding up a control circuit which implements a suspend/resume capability stipulated by the SDIO standard.

[0021]

The temporary memory can be used to store not only the data transferred and received between the SD interface and application interfaces, but also the data transferred and received among the application interfaces.

[0022]

Furthermore, the SDIO controller according to the present invention preferably comprises a microcontroller unit (MCU) for the data control.

[0023]

The use of the MCU makes it possible to help interpret SD commands received by the SD interface, to control any memory device connected to the memory interface, and to perform various processes such as applications, transmission of interrupt signals, preparation of transfer data, and debugging.

[0024]

In this case, the SDIO controller may further comprise an I/O (GPIO) to input and output control signals to/from the microcontroller unit (MCU).

[0025]

Incidentally, a GPIO, HS-UART, PCMCIA, etc. used to connect to input/output devices are referred to herein as function interfaces and they are differentiated from memory interfaces used to connect to memories which store and maintain data. The function interfaces and memory interfaces are referred to collectively as "application interfaces."

[0026]

The memory interfaces include an EEPROM, NAND-type flash memory, NOR-type flash memory, and other similar interfaces. The SDIO controller may comprise one or more of these interfaces or another type of memory interface.

[0027]

When equipped with a memory interface, the SDIO controller can be connected with a non-volatile memory (flash memory, ferroelectric memory, ferromagnetic memory, etc.) that is compatible with the memory interface.

[0028]

When connected with the non-volatile memory, the SDIO controller may comprise at least one of firmware, hardware information about the SDIO controller (CIS), driver software (CSA), and user data in part of memory areas.

[0029]

Also, the SDIO controller may comprise hardware information about application modules (firmware, CIS, and CSA) in part of the memory areas.

[0030]

A SDIO wireless communications module may be configured such that the SDIO controller according to the present invention will be connected with the wireless communication modules compliant with the communication standards, via the application interfaces.

[0031]

Examples of the communication standards include IEEE (Institute of Electrical and Electronic Engineers) 802.x, which, among others, may be IEEE802.11a/b/g/e.

[0032]

If the SDIO wireless communication module is put in the SDIO-compliant card, it becomes a wireless communication card. Alternatively, the module may be provided as it is so that it will be built into equipment which contains the SDIO host.

[0033]

Thus, the SDIO controller, according to the present invention, provides a useful tool for many developers who develop SDIO drivers, wireless communication modules, or other hardware, not to mention SDIO cards.

[0034]

[Embodiments of the Invention]

(Basic configuration of hardware)

Figure 1 shows basic configuration of the SDIO

controller according to the present invention, consisting of functional blocks. A representative configuration will be described with reference to an example. All functions are implemented by an LSI chip (e.g., a PBGA approximately 6 mm x 6 mm in size). An example of the SDIO controller according to the present invention comprises an I^2C EEPROM, PCMCIA socket (slot), and UART connector.

[0035]

As shown in Figure 1, the SDIO controller comprises a SD interface (HIM) 1 which deciphers commands received from a SDIO host and returns a necessary response to the SDIO host, a temporary (R/W FIFO) memory 2, and a plurality of application interfaces 3a, 3b, and 3c. In terms of functional blocks, a control register (SDIOREG) 4 is included in the HIM although they are shown separately. The control register 4 is required for the SDIO host to control SD devices.

[0036]

The application interfaces are an HS-UART (UART which supports higher speeds) 3a, a PCMCIA 3b, and a memory interface 3c to which a flash memory can be connected. Other application interfaces, such as those compatible with a PC card bus or with other interfaces, may also be used.

[0037]

The above functional blocks consist of minimum necessary components. However, this example further includes a FIFO controller (DMA) 5, a microcontroller unit (MCU) 6, I-RAM 7a, D-RAM 7b, and GPIO 8. Although one DMA

5 is shown in Figure 1, actually there are three DMAs. Each DMA is connected to RFIFOs 2a, 2b, and 2c which temporarily store data sent from DMAs for application interfaces as well as to a WFIFO 2d which temporarily store data sent from the SDIO host via the HIM. Functions of these components will be described.

[0038]

- Interfaces -

(1) SD interface

The SD interface interprets SD commands received from the SD host via the SD bus and returns a necessary response according to the SDIO standard. Depending on the commands, the SD interface checks the data sent via the SD bus for CRC error, stores it in the WFIFO, and reads pooled data out of the RFIFO and sends it with CRC error check code to the SD host via the SD bus.

[0039]

The SD bus consists of nine signals. Its functional and timing specifications have been established by the SDIO standard.

[0040]

When a SDIO card is inserted in the slot, the SDIO host recognizes card information and sends a large number of commands to identify recognize the inserted SDIO card. The HIM interprets the commands and returns a response to the host.

[0041]

After this exchange of data, when the SDIO card is

authorized, data from host devices (e.g., image data from a digital camera, image data from an IP phone, etc.) can be received and transmitted.

[0042]

(2) HS-UART interface

An HS-UART is a serial interface which is equipped with the temporary memory to support high-speed transfer. The HS-UART takes data out of the temporary memory, converts it into serial data, and transmits the data according to RS-232C standard. On the other hand, the HS-UART converts data received according to the RS-232C standard into parallel data, stores the resulting data in the temporary memory, and generates an interrupt to the SD host. The interface allows a modem, PHS, Bluetooth device, or other device with the RS-232C interface to be connected. The SD host controls the HS-UART and learns its status via SDREG.

[0043]

(3) PCMCIA interface

PCMCIA is an interface supported by notebook personal computers and the like. It is an international standard. Data is transferred between the temporary memory and PCMCIA devices according to the PCMCIA standard. Also, interrupts from PCMCIA devices are transmitted to the SD host via the SDREG and HIM. The interface allows an IEEE802.11x device, hard disk, compact disk, or other device with a PCMCIA interface to be connected.

[0044]

Incidentally, compact flashes (registered trademark)

and the like are basically a subset of the PCMCIA interface, and they are thus connected to the PCMCIA interface.

[0045]

(4) Memory interface

The memory interfaces according to the present invention include three types: EEPROM, NAND-type memory, and NOR-type memory. It is desirable that a development tool should have as many memory interfaces as possible.

[0046]

a. I^2C serial EEPROM interface

The interface takes data out of the temporary memory, converts it into serial data, and then writes it into a serial EEPROM according to the I^2C protocol. On the other hand, it reads out data from the serial EEPROM according to the I^2C protocol, converts it into parallel data, and stores it in the temporary memory. The interface allows the serial EEPROM to be connected.

b. NAND-type flash interface

The interface takes data out of the temporary memory and writes it into a NAND-type flash memory via a data line. On the other hand, it reads out data from the NAND-type flash memory via the data line and stores it in the temporary memory. Also, it generates read/write signals and address signals necessary for data reads and writes. The interface allows the NAND-type flash memory to be connected.

c. NOR-type flash interface

The interface takes data out of the temporary memory and writes it into a NOR-type flash memory via an IO line.

On the other hand, it reads data from the NOR-type flash memory via the IO line and stores it in the temporary memory. Also, it generates commands and addresses necessary for data reads and writes and sends them to the NOR-type flash memory via the IO line. The interface allows the NOR-type flash memory to be connected.

[0047]

Capabilities of memory cards can be added to a SDIO card using the memory interfaces described above.

[0048]

(5) GPIO (General Peripheral I/O) interface

The interface controls the direction and values of GPIO interface signals according to default values written by the SD host into the SDREG and sets the signal values in the SDREG. A GPIO interface in the input direction is used to make an interrupt request (IRQ) to the SD host as well as to make status notification of application devices such as Receive Ready. A GPIO interface in the output direction is used to control LEDs, to control power consumption of application devices, and to switch modes.

[0049]

(6) SDREG

The SDREG consists of a register defined by the SDIO standard and accessible from the SD host and a register for internal control. SDREG is also accessed from MCU. The description will be given to the register that is defined by the SDIO standard as user-dependent and the register for internal control.

[0050]

FN1 (HS-UART) register:

The HS-UART is equipped with a control register for controlling its operation. The control register can be seen directly from the host devices. The host devices can access the control register directly via the SDIO interface. Serial communications are conducted by the UART based on settings of the control register. In other words, by accessing the control register, the host devices can directly control data transmission and reception to/from various devices.

[0051]

A control register for the HS-UART interface is installed in an area of Function 1 defined by the SDIO standard. The SD host transmits and receives data to/from the HS-UART devices via this register. Specifically, a register equivalent to the one mounted on the 16650 chip from National Semiconductor Corporation.

[0052]

Specifically, the registers used include an RBR (read buffer register: holds receive data temporarily), THR (transmit holding register: holds transmit data temporarily), IER (interrupt enable register: used for interrupt control), IIR (interrupt identification register: indicates interrupt sources), REF (FIFO control register: controls transmit/receive FIFOs), LCR (line control register: controls data lines), MCR (modem control register: controls a modem), LSR (line status register:

displays data line status), MSR (modem status register: displays modem control line status), SCR (scratch register: general-purpose), DLL (divisor latch LS: divides a transmit/receive clock), DLM (divisor latch MS: divides a transmit/receive clock), and HFC (hardware flow control: controls a modem line by hardware).

[0053]

FN2 (PCMCIA) register:

A control register for the PCMCIA interface is installed in an area of Function 2 defined 0by the SDIO standard. The SD host transmits and receives data to/from PCMCIA devices via this register. Specifically, registers are installed to control data windows to PCMCIA attribute areas, data windows to memory areas, data windows to IO areas, address offset values of the data windows, and timings of PCMCIA signals as well as for interrupt enable control.

[0054]

Also, internal control registers are installed to store command identifiers and command arguments received from the SD host, specify data size for data transfer, indicate types of error occurring during command processing, and indicate SD bus mode as well as for the MCU to control the HIM, DMA, temporary memories, and application interfaces and check their status.

[0055]

(7) Temporary memories (WFIFO, RFIFO1, RFIFO2, RFIFO3)

These memories temporarily hold data to be transferred

between the SD interface and application interfaces or among the application interfaces. Herein, for the sake of convenience, the memory which temporarily holds data to be output to application interfaces is defined as a WFIFO and the memory which temporarily holds data received from application interfaces is defined as a RFIFO.

[0056]

One FIFO is enough in principle. However, one WFIFO and three RFIFOs are provided according to the present invention.

[0057]

The capacities of FIFO memories (RAMs) are as follows, for example:

RFIFO1... 512 bytes (for UART)

RFIFO2... 2 kB (for PCMCIA)

RFIFO3... 2 kB (for memories)

WFIFO ... 2 kB

[0058]

When there are a plurality of application interfaces, if an RFIFO is provided for each application interface in this way, even if data transfer from the first application interface to the SD host is temporally suspended, data can be transferred from the second application interface to the SD host using another RFIFO, and thus the transfer from the first application interface to its RFIFO can be continued. When the data transfer from the first application interface to the SD host is resumed, since data has been accumulated in the RFIFO dedicated to the first

application interface, the suspension does not have much effect on the data transfer rate. Also, the data received from application interfaces are pooled in the respective dedicated RFIFOs, the correspondence between data and application interfaces is explicit, making it easy to control the data.

[0059]

(8) DMA (direct memory access)

DMA is a control logic hardware. It transfers data between temporary memories and application interfaces. The DMA is controlled by the HIM or MCU. Only one DMA is shown in Figure 1 for the sake of convenience, but DMA is provided for each application interface and simultaneously transfers data between the application interfaces and FIFOs.

[0060]

- Microcontroller unit (MCU) -

The microcontroller unit (MCU) is a logic IC. It has the I-RAM 7a (8 kB) and D-RAM (256 kB). The microcontroller unit can control the GPIO, SDIOREG (FNO), and DMA as well as memory devices connected to the memory interfaces (M-IF). An 8-bit MCU (such as 80C51) will do. It has a 8-bit port, which transmits direction (input or output) signals and interrupt signals and performs wired-OR operations.

[0061]

Since there are only a few types of I/O command which the HIM receives from the SDIO host, I/O-only processing may be handled by the HIM alone without the MCU. However,

at least tens of commands are needed to control a flash memory mounted on a memory interface. In that case, the MCU can be used to control both I/O and flash interface. In addition, the MCU can control the memory devices, helping interpret SD commands received by the SD protocol engine (HIM), and performing various processes with respect to the applications, including transmission of interrupt signals, preparation of transfer data, and debugging.

[0062]

Next, operation of the SDIO controller will be described.

[0063]

1. The SD host sends a command. The command contains such information as the type of command, number of data items to be transferred, and destination.

2. The HIM interprets the command, and then returns a Command response to the SD host and generates an internal Command interrupt to the MCU.

3. In the case of a write command, data is transmitted when the SDIO host receives the response.

4. The HIM stores the data in the WFIFO. In the meantime, the interrupted MCU controls the DMA and application interfaces according to the content of the command and waits for the data from the SD host to arrive.

5. Upon receiving the data, the HIM returns a Data response to the SD host and generates an internal Data Ready interrupt to the MCU.

7. Upon receiving the Data Ready interrupt, the MCU starts

up the DMA.

8. The DMA is started and the data pooled in the WFIFO is transferred via a predetermined application interface.

[0064]

Incidentally, the HIM may start up the DMA by controlling the DMA and application interface directly and bypassing the MCU.

[0065]

(First embodiment) - SDIO controller equipped with a wireless communication module -

The wireless communication modules have been defined by IEEE (Institute of Electrical and Electronic Engineers) 802.x. For example, in the wireless LAN area, 802.11b modules are the mainstream at present and technology development is conducted aiming at IEEE802.11a/g/e or new standards with enhanced communications speed and security.

[0066]

Regarding wireless LAN cards compliant with IEEE802.11b, for example, a large number of products which are designed to connect to a PCMCIA bus are known at present. If one wants to use a wireless LAN card as a SD card, it is physically impossible to connect a 9-pin SDIO host with a 68-pin PCMCIA device directly.

[0067]

However, if protocol conversion into PCMCIA is carried out using the SDIO controller according to the present invention while satisfying SDIO specifications, a PCMCIA-compatible IEEE802.11b wireless front end can be

connected directly to the PCMCIA interface of the SDIO card. In other words, SDIO-compatible wireless LAN card can be developed at low costs in a short period of time using existing architectures and software as they are.

[0068]

[Embodiment]

(Example 1) SDIO wireless communication card

As shown in Figure 2, by putting the SDIO controller 9 and a wireless communication module 10 connected via a PCMCIA interface in a single card, it is possible to make a wireless communication card 11 which can be inserted in SDIO slots of host devices. Preferably, a IEEE802.x or other standardized wireless communication module is used.

[0069]

Figure 3 shows a SDIO controller 10 and IEEE802.11b wireless LAN module 20 contained in a single SDIO card. The broken lines indicate the IEEE802.11b wireless LAN module 20. The module consists of a medium access controller (MAC), base band processor (BBP) 22, and RF controller 23 and is connected to the SDIO controller 10 via a PCMCIA interface.

[0070]

When using a Bluetooth module as the wireless communications module, it can be connected to the SDIO controller according to the present invention via a HS-UART, a standard Bluetooth port. Since the SDIO controller according to the present invention can use existing application modules as SD applications, it can shorten the

labor and costs of design and development.

[0071]

Furthermore, since the SDIO controller according to the present invention is equipped with a plurality of application interfaces, if a plurality of wireless communication modules are installed, it allows selective use: for example, the user can connect to a wireless LAN when an access point is available and use PHS when no access point is available.

[0072]

(Variation) SDIO controller + wireless communication module + memory device

A flash memory 14a and/or EEPROM 14b may be installed on a memory interface 3c of the SDIO controller as shown in Figure 3. This will add capabilities of a memory card to the wireless LAN card. Of course, the wireless LAN card may be provided only for wireless LAN capabilities without the addition of memory card capabilities, for the purpose of cost reduction.

[0073]

In this way, by adapting an interface of an existing wireless communication module to an application interface of the SDIO controller, it is possible to develop applications of the SDIO controller at low costs in a short period of time.

[0074]

(Advantage of adding memory) - Commonality of firmware -

Installing a memory divide together with an application other than a memory device, such as a wireless communication module, has the advantage of achieving commonality of firmware and the like. This will be described below.

[0075]

Normally, an application (such as a wireless communication module) is provided with memory for recording firmware and the like. If the SDIO controller has a built-in memory device, firmware of the application and firmware and the like of the SDIO controller can be recorded in different parts of the same memory device. Beside firmware, other necessary information (CIS, CSA, etc.) may be stored.

[0076]

In that case, the firmware and the like of the SDIO controller are downloaded via a memory interface while the firmware and the like of the application are downloaded via a PCMCIA or other interfaces. It is advisable to store these boot programs in the firmware of the SDIO controller.

[0077]

(Example 2) SDIO wireless communication module

In the first example described above, the wireless card 25 itself contains a built-in wireless communication module (of course, memory or other applications may be added as required) as shown in Figure 4A. Today, PCs with built-in wireless LAN capabilities are available. They incorporate an IEEE802.11b wireless LAN module to be

connected to a PCMCIA card bus.

[0078]

To build the wireless communication LAN module into a device (e.g., digital video device, digital camera, PDA, etc.) other than a PC, the device must contain a built-in PCMCIA controller.

[0079]

However, building a PCMCIA controller into a mobile device such as a digital camera for wireless communication increases cost unnecessarily.

[0080]

On the other hand, digital cameras and other mobile devices which support SDIO cards essentially incorporate a SDIO host.

[0081]

Thus, as shown in Figure 4B, by integrating the SDIO controller according to the present invention and wireless communication capabilities on a single chip as a module and incorporating the resulting SDIO wireless communication module 26 into a host device, it is possible to build the wireless communications capabilities into the host device without incorporating a PCMCIA controller.

[0082]

Furthermore, if a SDIO slot 27 is provided by branching the SD bus, a SDIO card 28 can implement memory and other capabilities.

[0083]

When putting the memory and wireless communication

module in a single card, it is difficult to increase the memory capacity of the card. Besides, if a large-capacity memory-only card is prepared as well, it is necessary either to use both the memory card with the wireless communication capabilities and the large-capacity memory-only card by changing them or install two slots in the SD host device so that the two cards can be used simultaneously.

[0084]

On the other hand, if a wireless communication module is built into the device and the memory and other capabilities are incorporated into the SDIO card, this will be convenient for both the user and manufacturer because the end user can purchase peripherals further outside the SDIO slot by selecting necessary ones.

[0085]

Incidentally, since the SDIO controller according to the present invention has many application interfaces, if sockets, slots, or the like compatible with the application interfaces are provided, a compact flash (registered trademark) and other memory can be used through other interfaces such as a PCMCIA interface.

[0086]

(Other examples)

By mounting a wireless communication module on the SDIO host or inserting a SDIO card equipped with wireless communication capabilities, it is possible to use a wireless LAN in areas covered by access points, communicate with other devices in ad hoc mode, or implement cordless

telephone connection (IP phone) to the Internet.

[0087]

If various applications (memories, PCMCIA card slots, UART sockets, etc.) and development software are installed in the SDIO controller according to the present invention and thereby development environments are prepared, it becomes possible to provide certification services in relation to the internationally established SDIO standard.

[0088]

[Advantages of the Invention]

The present invention provides a highly versatile SDIO controller equipped with capabilities to interface a SDIO host with various applications (wireless, GPS, memory, etc.) as well as to provide other devices resulting from application of the SDIO controller. Furthermore, it enables coordinated development of I/O of storage media devices and software. For example, it can readily provide a compact, low-power wireless communication card compatible with SDIO or a module which can be mounted on the host devices.

[Brief Description of the Drawings]

[Figure 1]

Figure 1 is a diagram showing basic configuration of a SDIO controller according to the present invention, consisting of functional blocks;

[Figure 2]

Figure 2 is a diagram showing a wireless communications card consisting of a SDIO controller 9 and a wireless

communications module 10 connected via a PCMCIA interface and put in a single card;

[Figure 3]

Figure 3 is a diagram showing a SDIO controller 10 and IEEE802.11b wireless LAN module 20 contained in a single SDIO card;

[Figure 4]

Figure 4A is a diagram showing how a card 25 itself contains a built-in wireless communications module; and

Figure 4B is a diagram showing how a SDIO controller and wireless communications capabilities are integrated on a single chip as a module and incorporated into a host device.

[Description of Symbols]

1 SDIO host interface module (HIM)

2 Temporary memory (R/W FIFO)

3a, 3b, 3c Application interface

4 Control register (SDIOREG)

5 FIFO controller (DMA)

6 Microcontroller unit (MCU)

7a I-RAM

7b D-RAM

8 GPIO

9 SDIO controller

10 Wireless communications module

11 Wireless communications card which can be inserted in a SDIO slot

20 Wireless LAN module

22 Medium access controller (MAC) and base band processor (BBP)

23 RF controller

25 SDIO wireless card

26 SDIO wireless module

27 SDIO slot

28 SDIO card

Figure 1

#1 SDIO HOST

#2 APPLICATION 1

#3 APPLICATION 2

#4 APPLICATION 3 (MEMORY)

Figure 2

#1 DEVICE SIDE: SDIO HOST

11 SDIO CARD

9 SDIO CONTROLLER

Figure 4A

#1 SD BUS

#2 COMMUNICATIONS CAPABILITIES (+ MEMORY)

25 SD-Lik11x (+ MEMORY)

Figure 4B

#1 SD BUS

#2 MEMORY AND OTHER CAPABILITIES

28 SD MEMORY AND OTHER CAPABILITIES